DOI: 10.7763/IPEDR. 2013. V63. 15

# Average Concept of Crossover Operator in Real Coded Genetic Algorithm

Rosshairy Abd Rahman and Razamin Ramli+

Dept. Of Decision Science, School of Quantitative Sciences, Universiti Utara Malaysia

**Abstract.** As the most important search operator in a Genetic Algorithm (GA) approach, many procedures have been proposed to accomplish the idea of a crossover. As a result, knowledge in crossover has incorporated special features such as statistical elements (i.e. arithmetic crossover) and natural observation (i.e. queen bee crossover) to name a few. Thus, this paper proposed a mean or average concept of crossover for fitter parents to produce a new offspring in a GA based approach in an animal diet formulation problem. Experiments using real data were carried out involving GA models with average crossover and one-point crossover. Subsequently, the incorporation of power heuristics as a repair operator was investigated to find the best combination of ingredients, while removing the unwanted ones. Comparisons were made between GA models incorporating repair operator with different crossovers: average crossover and one point crossover. The results show that the performance of average crossover is comparable with that of the one-point crossover. The inclusion of the repair operator provides an advantage that shows interesting solution for the tested problem.

**Keywords:** Animal Diet Formulation, Crossover, Feed Mix Problem, Real Valued Genetic Algorithm.

## 1. Introduction

Genetic Algorithm (GA) is a search method based on natural evolution which has been introduced by Darwin. The process of GA starts by a selection of two individuals as parents from a set of possible solutions based on their fitness values. These parents then will go through several search operators such as crossover and mutation to produce a child or offspring. The process is repeated over a few generations while the high fitted individual with good characteristics will survived. At last, the best so far solution is discovered after going through all the procedures.

GA has several encoding types that are binary, real value and permutation [1]. The encoding type is normally chosen based on compatibility with the existing problem. Real value encoding is also known as real coded genetic algorithm (RCGA) [2]. Real number encoding using real values is introduced with the objective to bring GA search closer to the solution space of the problem. [3] suggested that, in developing optimization problems with continuous variables, real value is more appropriate to use since real number encoding provide more precision, faster and more consistent in each run compared to the binary encoding. Moreover, binary encoding requires much computing times especially in large problem domains. Hence, real value encoding is more appropriate to use in continuous domain [1][4][5].

Since the performance of GA highly relies on its search operators, many studies have been proposed to enhance the performance of GA with respect to its search operators, especially the crossover [6]. Ideas and innovations in crossover include special features such as statistical elements (i.e. arithmetic crossover) [3] and natural observation (i.e. queen bee crossover) [7] to name a few. As a result, in this paper, we introduce an average crossover for a continuous problem using real value encoding where the chromosome may adopt floating point value.

## 2. Methodology

Average crossover is a modification of the traditional single-point crossover. In relation to the average crossover, the mean values from both fitness values of parents are calculated. Then, a midpoint is selected as a crossover point. The crossover process is carried out using the average value of both parents. Fig. 1 illustrates how this reproduction process occurs between both parents.

+ Corresponding author. Tel.: + 6049286968; fax: +6049286906.
*E-mail address*: shairy@uum.edu.my.

Parent 1

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 6 | 28 | 3 | 40 | 37 | 6 | 8 | 3 |

Parent 2

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 5 | 16 | 4 | 7 | 30 | 14 | 14 | 3 |

Child 1

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 5.5 | 22 | 3.5 | 23.5 | 37 | 6 | 8 | 3 |

Child 2

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 6 | 28 | 3 | 40 | 33.5 | 10 | 11 | 3 |

Fig. 1: The illustration of average crossover

This new crossover is tested using animal diet formulation problem, specifically for shrimp. The diet consists of a string of ingredients with respected nutrients. This string is also known as chromosome. The aim of this crossover is to fully utilize the fitter parents to produce a new improved string of diet composition in terms of minimizing cost. Average crossover is used in conjunction with roulette wheel selection and power mutation as introduced by [8]. Instead of random initialization of the population, a repair operator named Power Heuristics is introduced in the initial phase to cater for the strictness of nutrient requirements and ingredients restriction. We adapted Power Heuristics from power mutation [8] to satisfy nutrients' constraints and at the same time, able to remove inappropriate ingredients when a nutrient restriction is violated. The incorporation of this Power Heuristics is able to reduce the initial penalty function values, thus increases the opportunity of acquiring a feasible solution. Please refer to [9] for further explanation on Power Heuristics. Overall, four GA models are tested; one-point crossover without Power Heuristics, average crossover without Power Heuristics, one-point crossover with Power Heuristics, and average crossover with Power Heuristics. Generally, the algorithm steps are as follows:

- Initialization
- Roulette wheel selection
- Average crossover or one-point crossover
- Power mutation
- Power Heuristics
- Steady state reproduction

Elitism procedure [10] is also used in this GA such that some best found chromosomes or solutions can stay on to the next generation. The whole process is repeated and continued until a predetermined number of generations is completed or a stopping criterion is met. Summary of the operators used in the four GA models is exhibited in Table 1.

Table 1: Summary of the operators used in the four GA models

| Operators | GA models | | | |
|---|---|---|---|---|
| | *one point* | *average* | *'ower Heuristic one point* | *Power Heuristics Average* |
| Initialization | No Power Heuristics | No Power Heuristic | Power Heuristics | Power Heuristics |
| Selection | Roulette | Roulette | Roulette | Roulette |
| Crossover | One-point | Average | One-point | Average |
| Mutation | Power mutatio | Power mutatio | Power mutatio Power Heurist | Power mutatior Power Heuristi |

## 3. Test Problem

The performance of four GA models is tested using real data for animal diet formulation problem. In this problem, the aim is to satisfy all the nutritional needs of farmed shrimps at a minimum cost. The minimization problem takes into account 14 ingredients and 18 nutrients. The following are the objective function and constraints involved in this problem. Objective function of the feed cost is defined as:

$$f(s) = \min \sum^{n} (X_i \times C_i)$$

where $Ci$ is the cost of ingredient $i$, and $Xi$ equals the weight of the $i$th ingredient. $s$ is cumulative cost in a string of chromosome

However, the aim of this study is to firstly reduce the penalty function value based on all identified constraints. The constraints consist of ingredients' range, ingredient (ration) weight, number of ingredients, single nutrient's range, combination nutrients' range and ratio of nutrients.

- Ingredients' range

  $X_i = 0$ or $L_{Xi} \leq X_i \leq U_{Xi}$ for all $X_i$

where $L_{Xi}$ = lower bound of ingredient $i$, $U_{Xi}$ = upper bound of ingredient $i$, $Xi$ equals the weight of the $i$th ingredient.

- Ingredient weight

$$\sum_{i=1}^{n} X_i = Y$$

Y is a weight predefine by user in user interface

- Number of ingredient

$$n \leq 14$$

- Single nutrients' range

$$L_{Nk} \leq \sum_{i=1}^{n} N_{ki} X_i \leq U_{Nk}$$

where $L_{Nk}$ = lower bound of nutrient $k$, $U_{Nk}$ = upper bound of nutrient $k$, $N$ = Total value of nutrient $k$

- Combination nutrients' range

$$L_{Nk(i+j)} \leq \sum_{i=1}^{n} N_{k(i+j)} X_i \leq U_{Nk(i+j)}$$

where $L_{Nk(i+j)}$ = lower bound of combination nutrient $i+j$, $U_{Nk(i+j)}$ = upper bound of combination nutrient $i+j$

- Ratio nutrients' range

$$L_{ratio} \leq \frac{\sum_{i=1}^{n} N_{ki}}{\sum_{i=1}^{n} N_{kj}} \leq U_{ratio}$$

where $L_{ratio}$ = lower bound of ratio between nutrient $i$ and $j$, $U_{ratio}$ = upper bound of ratio between nutrient $i$ and $j$

Fitness calculation for the GA is basically based on penalty value for each constraint. There are two types of constraint existed; hard and soft constraints. In this study, hard constraints are ingredient (ration) weight, number of ingredient and protein range constraint i.e. $N_1$. Else, for soft constraints, different penalty values are given for different constraints based on in depth discussion with experts. Penalty value of 20 is given for violating each ingredient constraint, 30 for single nutrient, 20 for combination of nutrients and 20 for ratio of nutrient. The list of ingredients with its restriction can be viewed in Table III. Later, Table IV shows the list of nutrients and its restriction which need to be fulfilled.

# 4. Result and discussion

In our experiments, GA parameters were set as follow: size of a population is 30, number of generation is 100, crossover rate is 0.60, and power value for power mutation is 0.25. Table II illustrates the simulated results of all GA models. From the Table, we summarize the average fitness and processing time (in second) taken to produce the best-so-far solution. These two values are used as an indicator to evaluate the performance of these GA models.

Table 2: The results obtained from GA models

| GA models | Average fitness | Time Taken |
|---|---|---|
| GA-one point | All solution Infeasible | 2038.5 |
| GA-average | All solution Infeasible | 2043.2 |
| GA repair- one point | 6/30 infeasible 557.5 | 2548.2 |
| GA repair- average | 2/30 infeasible 571.786 | 2295.833 |

From Table II, both GA models using one-point crossover and average crossover give infeasible solutions. In 30 runs of each set of generations, the best average fitness is achieved from GA Power Heuristics -one point model. However, out of these 30 runs, six infeasible solutions were obtained. The GA Power Heuristics-average model produced slightly higher fitness value, but with only two infeasible solutions. In addition, the average time taken by this model is faster than that of the GA Power Heuristics-one point crossover.

The incorporation of Power Heuristics gives relatively big difference in results if compared to the case with no inclusion of Power Heuristics. It shows that Power Heuristics plays important roles of finding a better combination of ingredient in the animal diet formulation problem. The results from GA Power Heuristics-average in Table III show that, only eight ingredients are selected, while the six others equal to zero. Total weight for the solution is 100kg, and the penalty value is 350. Total cost for this feedstock is MYR217.081. All ingredients are in the acceptable range, except for the last ingredient ($X_{14}$) that exceed the allowable value of 0.236. The benefit of adding the Power Heuristics to the GA is that one can include many ingredients in the dataset to be selected in the diet model. This action may introduce new potential ingredients without deleting any existing ingredients.

Several nutrients are in the acceptable range, while the rest are outside the range. The result obtains for ingredients and nutrients from GA Power Heuristics-average model can be viewed as well in the following Table III and Table IV.

## 5. Conclusion

Table 3: The result for ingredients and its restriction

| Ingredien | Minimum | Maximum | Quantity |
|---|---|---|---|
| $X_1$ | 5 | 10 | 0 |
| $X_2$ | 15 | 50 | 42.586 |
| $X_3$ | 3 | 5 | 4.043 |
| $X_4$ | 5 | 50 | 0 |
| $X_5$ | 30 | 40 | 0 |
| $X_6$ | 5 | 15 | 14.303 |
| $X_7$ | 5 | 15 | 13.113 |
| $X_8$ | 2 | 5 | 4.023 |
| $X_9$ | 15 | 60 | 0 |
| $X_{10}$ | 5 | 15 | 0 |
| $X_{11}$ | 5 | 15 | 12.135 |
| $X^{12}$ | 3 | 5 | 0 |
| $X^{13}$ | 3 | 5 | 4.612 |
| $X_{14}$ | 3 | 5 | 5.236 |

The performance of basic GA model with Power Heuristics is described. Much work is needed which could include other beneficial steps in the model. In this paper, two new features are introduced; average crossover and Power Heuristics. Four GA models are tested with different operators. The results show that our proposed average crossover works well in real value encoding. Other crossover types including multipoint and uniform crossovers can also be injected with average feature. The incorporation of Power Heuristics is good at finding better solution even though relatively more time is required. Power Heuristics as a repair operator can be generalized to other similar problems such as blending problems, where many ingredients can be considered.

Table 4: The results for nutrients and its restriction

| Nutrient | Minimum (%) | Maximum (%) | Quantity (%) |
|---|---|---|---|
| $N_1$ | 38 | 45 | 44.139 |
| $N_2$ | 0.08 | 0.18 | 8.541 |
| $N_3$ | 0 | 15 | 8.095 |
| $N_4$ | 0.3 | 0.7 | 4.665 |
| $N_5$ | 0 | 4 | 0.663 |
| $N_6$ | 0 | 2.3 | 0.504 |
| $N_7$ | 2.2 | 2.32 | 2.831 |
| $N_8$ | 0.6 | 0.84 | 0.964 |
| $N_9$ | 1 | 1.33 | 6.032 |
| $N_{10}$ | 1.7 | 2.16 | 7.520 |
| $N_{11}$ | 1.55 | 1.65 | 2.622 |
| $N_{12}$ | 0.7 | 0.96 | 0.726 |
| $N_{13}$ | 1.4 | 1.6 | 1.954 |
| $N_{14}$ | 1.3 | 1.44 | 0.433 |
| $N_{15}$ | 0.2 | 0.32 | 1.609 |
| $N_{16}$ | 1.4 | 1.6 | 2.123 |
| $N_{17}$ | 0 | 1000 | 0.937 |
| $N_{18}$ | 0 | 1000 | 0.603 |
| $N_{12}+N_{18}$ | 1 | 1.44 | 1.329 |
| $N_{13}+N_{17}$ | 2.7 | 7.1 | 2.892 |
| $N_4:N_5$ | 1 | 1.3 | 1.312 |

# 6. References


[1] K. Deb, Introduction to representations. In: T. Bäck, D. B. Fogel, & T. Michalewicz, Evolutionary Computation 1: Basic Algorithms and Operators (pp. 127-131). Bristol: Institute of Physics Publishing, 2000.

[2] M. A. Şahman, M. Çunkaş, Ş. İnal, F. İnal, B. Coşkun, and U. Taşkiran, Cost optimization of feed mixes by genetic algorithms. *Advances in Engineering Software*. 2009, **40**, 965-974.

[3] Z. Michalewicz, *Genetic Algorithms+Data Structures=Evolution Programs*. (2nd, Ed.), New York: Springer-Verlag Berlin Heidelberg, 1994.

[4] S. K. Sung, K. Il-Hwan, V. Mani, and J. K. Hyung, Real-coded genetic algorithm for machining condition optimization, *Int J Adv Manuf Technol*. 2008, **38**, 884-895.

[5] K. Deep, and M. Thakur, A new mutation operator for real coded genetic algorithms. *Applied Mathematics and Computation*. 2007, **193**, 211-230.

[6] F. Herrera, M. Lozano, and A. M. Sánchez, Hybrid crossover operators for real-coded genetic algorithms: an experimental study. *Soft Comput*. 2005, **9**, 280-298.

[7] M. F. Azeem, and A. M. Saad, Modified queen bee evolution based genetic algorithm for tuning of scaling factors of fuzzy knowledge base controller. *IEEE Indicon 2004*, 299-303. Kharagpur: Indian Institute of Technology.

[8] K. Deep, K. Singh, M. L. Kansal, and C. Mohan, A real coded genetic algorithm for solving integer and mixed integer optimization problems. *Applied Mathematics and Computation*. 2009, **212**, 505-518.

[9] R. A. Rahman, and Ramli, R, A Hybrid Procedure in an Evolutionary Algorithm for an Animal Diet Problem. Proceeding of the International Conference of Engineering & Applied Science (ICEAS 2012). July 24-27, 2012, Beijing, China.

[10] D. E. Goldberg, *Genetic Algorithms in search, optimization, and machine learning*. Canada: Addison-Wesley Publishing Company, Inc., 1989, pp. 1-379.